MASSACHUSETTS INSTITUTE OF TECHNOLOGY VLSI PUBLICATIONS

VLSI Memo No. 88-480
October 1988

A MIXED FREQUENCY-TIME APPROACH FOR DISTORTION ANALYSIS OF SWITCHING FILTER CIRCUITS

K. Kundert, J. White, A. Sangiovanni-Vincentelli

Abstract

Designers of switching filter circuits are often interested in steady-state and intermodulation distortion due to both static effects, such as nonlinearities in the capacitors, and dynamic effects, such as the charge injection during MOS transistor switching or slow operational amplifier settling. Steady-state distortion can be computed using the circuit simulation program SPICE, but this approach is computationally very expensive. Specialized programs for switched capacitor filters can be used to rapidly compute steady-state distortion, but do not consider dynamic effects. In this paper we present a new mixed frequency-time approach for computing both steady-state and intermodulation distortion. The method is both computationally efficient and includes both static and dynamic distortion sources. The method has been implemented in a C program, *Nitswit*, and results from several examples are presented.

Microsystems Research Center Room 39-321
Massachusetts Institute of Technology
Cambridge Massachusetts 02139
Telephone (617) 253-8138

A Mixed Frequency-Time Approach for Distortion Analysis of Switching Filter Circuits

K. Kundert, J. White, A. Sangiovanni-Vincentelli
Dept. of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

Designers of switching filter circuits are often interested in steady-state and intermodulation distortion due to both static effects, such as nonlinearities in the capacitors, and dynamic effects, such as the charge injection during MOS transistor switching or slow operational amplifier settling. Steady-state distortion can be computed using the circuit simulation program SPICE, but this approach is computationally very expensive. Specialized programs for switched capacitor filters can be used to rapidly compute steady-state distortion, but do not consider dynamic effects. In this paper we present a new mixed frequency-time approach for computing both steady-state and intermodulation distortion. The method is both computationally efficient and includes both static and dynamic distortion sources. The method has been implemented in a C program, *Nitswit*, and results from several examples are presented.

1 Introduction

In general, analog circuit designers rely heavily on circuit simulation programs like SPICE [nagel75] or ASTAP [weeks73] to insure the correctness and the performance of their designs. These programs simulate a circuit by first constructing a system of differential equations that describes the circuit, and then solving the system numerically with a time discretization method such as backward-Euler. When applied to simulating switching filter circuits, such as the switched-

capacitor filters used in integrated circuits or the switching converters used in high power applications, the classical circuit simulation algorithms become extraordinarily computationally expensive. This is because the period of the clock is usually orders of magnitude smaller than the time interval of interest to a designer. The nature of the calculations used in a circuit simulator implies that an accurate solution must be computed for every cycle of the clock in the interval of interest, and this can involve thousands of cycles.

The most common approach to reducing the computational burden of switching filter simulation is to first to break the circuit up into functional blocks such as operational amplifiers and switches. Each functional bock is simulated, using a traditional circuit simulator, for some short period. The simulations of the functional blocks are used to construct extremely simple macromodels, which replace the functional blocks in the circuit. The result is a much simplified circuit that can be simulated easily. This simplified circuit is then simulated for the thousands of clocks cycles necessary to construct a solution meaningful enough to verify the design.

In programs specifically for switched-capacitor filters, like *Diana* [deman80] and *Switcap* [tsividis83], the simulation efficiency is enormously increased by the use of the "slow-clock" approximation. After each clock transition, every node in the circuit is assumed to reach its equilibrium point before another transition occurs. This assumption, along with the use of algebraic macromodels, allow the filter to be treated as a discrete-time system with one time point per clock transition. A set of difference equations is then used to describe the filter.

Specialized simulation programs are extremely efficient for determining frequency and time domain response of switching filters, but macromodels in general, as well as the "slow clock" approximation, tend to ignore second-order

effects that can change distortion characteristics. In particular, as switching filters are being pushed to operate at ever higher frequencies, the assumption that signals reach equilibrium between clock transitions is often violated. Also, since signals between clock transitions are not computed, it is possible to miss events that occur in these intervals that might interfere with proper operation and contribute to distortion (e.g., clock feed-through spikes causing an operational amplifier to saturate). Lastly, it is not possible to capture the effects of dynamic distortion processes, such as the important effect of the channel conductance on charge redistribution when a transistor switch turns off.

In this paper we present another approach to the simulation of switching filter circuits that is very efficient for calculating steady-state or intermodulation distortion, but does not depend on macromodels or the slow clock approximation. The method exploits the property of switching filter circuits that node voltage waveforms over a given high frequency clock cycle are similar, but not exact duplicates, of the node voltages waveforms in proceeding or following cycles. This suggests that it is possible to construct a solution accurate over many high frequency clock cycles by calculating the solution accurately for a few selected cycles.

In the next section we begin by describing our assumptions about switching filter circuits and presenting the mixed frequency-time method. In Section 3, we discuss some of the computations involved in the method. In Section 4 we briefly describe our program, *Nitswit*, and present comparison and application results. Finally, in Section 5, we present our conclusions and acknowledgements.

2 The Mixed Frequency-Time Method

Very little can be assumed about the behavior of the node voltage waveforms in a switching filter circuit over a given clock cycle, because the circuits involved are very nonlinear and are usually switching rapidly. However, the node voltage waveforms over a whole clock cycle usually vary slowly from one cycle to the next, as controlled by the input signal. This implies that if the input is periodic, and the switching filter circuit is in steady-state, then the sequence formed by sampling the node voltages at the beginning of each clock cycle is periodic (Fig. 1). We derive our method by assuming this to be true, and further assuming that the periodic function that describes the sequence of initial points in each clock cycle can be accurately represented as a truncated Fourier series using few terms.

If the sequence of initial points of each clock cycle can be described by a Fourier series with J terms, then once J initial points are known, all the initial points are known. This implies that given our Fourier assumption, to compute the steady state behavior of a switching filter circuit we need only find the initial points of J clock cycles (a similar idea in a different context was presented in [chua81]).

In the next two subsections we describe two relationships that can be exploited to construct a nonlinear algebraic system of J equations in J initial points (solving this system is discussed in Section 3). The first relation, described in section 2.1, is derived from the Fourier series assumption, and is a linear relationship between the initial points of an evenly distributed set of J cycles and the initial points of the corresponding J cycles that immediately follow (Fig. 2). The second relation is derived from solving the differential equation system that describes the analog circuit, for the time interval of one clock cycle,

J times, each time using one of the distributed set of J initial points as an initial condition. This results in another set of values for the initial points of the following J cycles. Insisting that this set match the set resulting from the Fourier relation yields a nonlinear algebraic system in J unknowns, which can be solved for the J initial points, and this is described in section 2.2.

2.1 The Delay Operator

Consider the sequence of initial points of each clock cycle at some circuit node n, and denote the sequence by $v_n(\tau_1), v_n(\tau_2), \ldots, v_n(\tau_S)$ where S is the number of clock cycles in an input period (Fig. 1). If it is assumed that this sequence can be accurately approximated by a truncated Fourier series, then

$$v_n(\tau_s) = V_0 + \sum_{k=1}^{K}(V_k^C \cos k\omega\tau_s + V_k^S \sin k\omega\tau_s), \tag{1}$$

where ω is the fundamental frequency of the input signal, K is the number of harmonics and $J = 2K + 1$ is the number of unknown coefficients. Given (1), there is a linear relation between any collection of J initial points and any other collection of J initial points. However, as mentioned above, we are most interested in the linear operator that maps a collection $v(\tau_{\eta_1}), \ldots, v(\tau_{\eta_J})$ into $v(\tau_{\eta_1} + T), \ldots, v(\tau_{\eta_J} + T)$ where T is the clock period and $\{\eta_1, \ldots, \eta_J\}$ is a subset of $\{1, \ldots, S\}$ (Fig. 2). This linear operator will be referred to as the delay matrix.

Deriving the delay matrix is a two stage process. First, the J points $v(\tau_{\eta_1}), \ldots, v(\tau_{\eta_J})$ are used to calculate the Fourier coefficients. Then the Fourier series (using these coefficients) is evaluated at the J times, $\tau_{\eta_1} + T, \ldots, \tau_{\eta_J} + T$. The Fourier coefficients are then eliminated to yield the desired direct relation. To compute the Fourier coefficients, write (1) as a system of J linear equation in J unknowns

[kundert88a].

$$\Gamma^{-1}\begin{bmatrix} V_0 \\ V_1^C \\ V_1^S \\ \vdots \\ V_K^C \\ V_K^S \end{bmatrix} = \begin{bmatrix} v_n(\tau_{\eta_1}) \\ v_n(\tau_{\eta_2}) \\ v_n(\tau_{\eta_3}) \\ \vdots \\ v_n(\tau_{\eta_J}) \end{bmatrix} \tag{2}$$

where $\Gamma^{-1} \in \Re^{J\times J}$ is given by

$$\begin{bmatrix} 1 & \cos\omega\tau_{\eta_1} & \sin\omega\tau_{\eta_1} & \cdots & \cos K\omega\tau_{\eta_1} & \sin K\omega\tau_{\eta_1} \\ 1 & \cos\omega\tau_{\eta_2} & \sin\omega\tau_{\eta_2} & \cdots & \cos K\omega\tau_{\eta_2} & \sin K\omega\tau_{\eta_2} \\ 1 & \cos\omega\tau_{\eta_3} & \sin\omega\tau_{\eta_3} & \cdots & \cos K\omega\tau_{\eta_3} & \sin K\omega\tau_{\eta_3} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos\omega\tau_{\eta_J} & \sin\omega\tau_{\eta_J} & \cdots & \cos K\omega\tau_{\eta_J} & \sin K\omega\tau_{\eta_J} \end{bmatrix}. \tag{3}$$

The matrix Γ^{-1} maps the Fourier coefficients to a sequence and is referred to as the inverse discrete Fourier transform. If the times $\tau_{\eta_1}, \ldots, \tau_{\eta_J}$ are reasonably evenly distributed over one period of the input signal, then Γ^{-1} is invertible. Its inverse, the forward discrete Fourier transform, is denoted by Γ. We can also write

$$\Gamma^{-1}(T)\begin{bmatrix} V_0 \\ V_1^C \\ V_1^S \\ \vdots \\ V_K^C \\ V_K^S \end{bmatrix} = \begin{bmatrix} v_n(\tau_{\eta_1}+T) \\ v_n(\tau_{\eta_2}+T) \\ v_n(\tau_{\eta_3}+T) \\ \vdots \\ v_n(\tau_{\eta_J}+T) \end{bmatrix}, \tag{4}$$

where $\Gamma^{-1}(T) \in \Re^{J\times J}$ is given by

$$\begin{bmatrix} 1 & \cos\omega(\tau_{\eta_1}+T) & \sin\omega(\tau_{\eta_1}+T) & \cdots & \sin K\omega(\tau_{\eta_1}+T) \\ 1 & \cos\omega(\tau_{\eta_2}+T) & \sin\omega(\tau_{\eta_2}+T) & \cdots & \sin K\omega(\tau_{\eta_2}+T) \\ 1 & \cos\omega(\tau_{\eta_3}+T) & \sin\omega(\tau_{\eta_3}+T) & \cdots & \sin K\omega(\tau_{\eta_3}+T) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \cos\omega(\tau_{\eta_J}+T) & \sin\omega(\tau_{\eta_J}+T) & \cdots & \sin K\omega(\tau_{\eta_J}+T) \end{bmatrix}. \tag{5}$$

Given a sequence, a delayed version is computed by applying Γ to the sequence to compute the Fourier coefficients, and then multiplying the vector of coefficients

by $\Gamma^{-1}(T)$.

$$\begin{bmatrix} v_n(\tau_{\eta_1}+T) \\ v_n(\tau_{\eta_2}+T) \\ \vdots \\ v_n(\tau_{\eta_J}+T) \end{bmatrix} = \Gamma^{-1}(T)\Gamma \begin{bmatrix} v_n(\tau_{\eta_1}) \\ v_n(\tau_{\eta_2}) \\ \vdots \\ v_n(\tau_{\eta_J}) \end{bmatrix}. \tag{6}$$

Thus, the delay matrix, $\mathcal{D}(T) \in \Re^{J\times J}$, is defined as

$$\mathcal{D}(T) = \Gamma^{-1}(T)\Gamma. \tag{7}$$

As the delay matrix is a function only of ω, K, $\{\tau_{\eta_1},\ldots,\tau_{\eta_J}\}$ and T, it can be computed once and used for every node.

In this derivation it was assumed that the sequence of cycle initial points is well represented by a truncated Fourier series of a single fundamental. If intermodulation distortion is to be calculated, the input will be the sum of two frequencies, and in that case the cycle initial points will be described by a generalized Fourier series with more than one fundamental frequency [kundert88]. The derivation of the delay operator for this generalized case is a straightforward extension of the steps used above. The delay operator so generated will still relate $v(\tau_{\eta_1}),\ldots,v(\tau_{\eta_J})$ to $v(\tau_{\eta_1}+T),\ldots,v(\tau_{\eta_J}+T)$, but J will be equal to the total number of terms in the generalized Fourier series.

2.2 The Differential Equation Relation

We assume that any switching filter circuit to be simulated can be described by a system of differential equations of the form

$$\frac{d}{dt}q(v(t),u(t)) + i(v(t),u(t)) = 0, \tag{8}$$

where $v(t) \in \Re^N$ is the vector of node voltages, $u(t) \in \Re^M$ is the vector of input sources, $q(v(t),u(t)) \in \Re^N$ is the vector of sums of charges at each node, and $i(v(t),u(t)) \in \Re^N$ is the vector of sums of currents entering each node. If the

node voltages are known at some time t_0, it is possible to solve (8) and compute the node voltages at some later time t_1. In general, one can write

$$v(t_1) = \phi(v(t_0), t_0, t_1) \tag{9}$$

where ϕ is referred to as the state transition function for the differential equation and can be expanded as

$$\phi(v(t_0), t_0, t_1) = \begin{bmatrix} \phi_1(v(t_0), t_0, t_1) \\ \vdots \\ \phi_N(v(t_0), t_0, t_1) \end{bmatrix} \tag{10}$$

where $\phi_n : \Re^{N\times 1\times 1} \rightarrow \Re$ for all $n \in \{1, \ldots, N\}$.

Now reconsider the J initial points at some circuit node n, $v_n(\tau_{\eta_1}), \ldots, v_n(\tau_{\eta_J})$ (Fig 2). For each $j \in \{1, \ldots, J\}$ and each $n \in \{1, \ldots, N\}$ we can write

$$v_n(\tau_{\eta_j} + T) = \phi_n(v(\tau_{\eta_j}), \tau_{\eta_j}, \tau_{\eta_j} + T) \tag{11}$$

where T is the clock period. Note that $v_n(\tau_{\eta_j} + T)$ is the initial point of the cycle immediately following the cycle beginning at τ_{η_j}. Also, the node voltages at τ_{η_j} can be related to the node voltages at $\tau_{\eta_j} + T$ by the delay matrix, $\mathcal{D}(T)$. That is,

$$\mathcal{D}(T) \begin{bmatrix} v_n(\tau_{\eta_1}) \\ \vdots \\ v_n(\tau_{\eta_J}) \end{bmatrix} = \begin{bmatrix} v_n(\tau_{\eta_1} + T) \\ \vdots \\ v_n(\tau_{\eta_J} + T) \end{bmatrix}. \tag{12}$$

It is possible to use (11) to eliminate the $v_n(\tau_{\eta_j} + T)$ terms from (12), which yields

$$\mathcal{D}(T) \begin{bmatrix} v_n(\tau_{\eta_1}) \\ \vdots \\ v_n(\tau_{\eta_J}) \end{bmatrix} = \begin{bmatrix} \phi_n(v(\tau_{\eta_1}), \tau_{\eta_1}, \tau_{\eta_1} + T) \\ \vdots \\ \phi_n(v(\tau_{\eta_J}), \tau_{\eta_J}, \tau_{\eta_J} + T) \end{bmatrix} \tag{13}$$

for each $n \in \{1, \ldots, N\}$.

3 Solution by Newton-Raphson

The collection of equations given in (13) can be reorganized into a system of NJ equations in NJ unknowns as

$$F\begin{pmatrix} v(\tau_{\eta_1}) \\ \vdots \\ v(\tau_{\eta_J}) \end{pmatrix} = \mathcal{D}_N(T)\begin{bmatrix} v_1(\tau_{\eta_1}) \\ \vdots \\ v_N(\tau_{\eta_1}) \\ \vdots \\ v_1(\tau_{\eta_J}) \\ \vdots \\ v_N(\tau_{\eta_J}) \end{bmatrix} - \begin{bmatrix} \phi_1(v(\tau_{\eta_1}), \tau_{\eta_1}, \tau_{\eta_1}+T) \\ \vdots \\ \phi_N(v(\tau_{\eta_1}), \tau_{\eta_1}, \tau_{\eta_1}+T) \\ \vdots \\ \phi_1(v(\tau_{\eta_J}), \tau_{\eta_J}, \tau_{\eta_J}+T) \\ \vdots \\ \phi_N(v(\tau_{\eta_J}), \tau_{\eta_J}, \tau_{\eta_J}+T) \end{bmatrix} \tag{14}$$

and

$$F\begin{pmatrix} v(\tau_{\eta_1}) \\ \vdots \\ v(\tau_{\eta_J}) \end{pmatrix} = 0, \tag{15}$$

where $F : \Re^{NJ} \to \Re^{NJ}$, and $\mathcal{D}_N \in \Re^{NJ \times NJ}$ is given by

$$\mathcal{D}_N(T) = \begin{bmatrix} d_{11}I_N & \dots & d_{1J}I_N \\ \vdots & & \vdots \\ d_{J1}I_N & \dots & d_{JJ}I_N \end{bmatrix} \tag{16}$$

where $d_{ij} \in \Re$ is the ij^{th} element of the delay matrix $\mathcal{D}(T)$, and $I_N \in \Re^N$ is the identity matrix.

Applying Newton's method to (14) leads to the iteration equation

$$J_F\begin{pmatrix} v^{(l)}(\tau_{\eta_1}) \\ \vdots \\ v^{(l)}(\tau_{\eta_J}) \end{pmatrix}\begin{bmatrix} v^{(l+1)}(\tau_{\eta_1}) - v^{(l)}(\tau_{\eta_1}) \\ \vdots \\ v^{(l+1)}(\tau_{\eta_J}) - v^{(l)}(\tau_{\eta_J}) \end{bmatrix} = -F\begin{pmatrix} v^{(l)}(\tau_{\eta_1}) \\ \vdots \\ v^{(l)}(\tau_{\eta_J}) \end{pmatrix} \tag{17}$$

where l is the iteration number and $J_F \in \Re^{NJ \times NJ}$ is the Frechet derivative of F given by

$$\mathcal{D}_N(T) - \operatorname{diag}\left(\frac{\partial\phi(v(\tau_{\eta_1}), \tau_{\eta_1}, \tau_{\eta_1}+T)}{\partial v(t_{\eta_1})}, \dots, \frac{\partial\phi(v(\tau_{\eta_J}), \tau_{\eta_J}, \tau_{\eta_J}+T)}{\partial v(t_{\eta_J})}\right). \tag{18}$$

There are two important pieces to the computation of one Newton iteration: factoring the matrix J_F, which is sparse, and evaluating J_F and F, which involves computing the state transition function, $\phi(v(\tau_{\eta_j}), \tau_{\eta_j}, \tau_{\eta_j}+T)$, and its

derivative for each $j \in \{1, \ldots, J\}$. The state transition functions can be evaluated by numerically integrating (8) over the J periods. The derivatives of the state transition functions, referred to as the sensitivity matrices, can be computed with a small amount of additional work during the numerical integration[trick75].

To show how the computation of the state transition function and its derivative fit together, consider numerically integrating (8) with backward-Euler, which we chose for simplicity and because it appears to be one of the best formulas for switching filter circuits. Given some initial time t_0 and some initial condition, $v(t_0)$, applying backward-Euler to (8) results in the following algebraic equation,

$$f(v(t_0+h), v(t_0)) = \frac{1}{h}(q(v(t_0+h)) - q(v(t_0))) + i(v(t_0+h)) = 0 \qquad (19)$$

where $h \in \Re$ is the timestep. Note we have dropped explicitly denoting the dependence of q and i on the input u for simplicity.

Equation (19) is usually solved with Newton-Raphson, for which the iteration equation is

$$J_f(v^{(l)}(t_0+h))(v^{(l+1)}(t_0+h) - v^{(l)}(t_0+h)) = -f(v^{(l)}(t_0+h), v^{(l)}(t_0)) \quad (20)$$

where $J_f(v(t)) \in \Re^{N \times N}$ is the Frechet derivative of the nonlinear equation in (19) and is given by

$$J_f(v(t)) = \frac{\partial f(v(t), v(t_0))}{\partial v(t)} = \frac{1}{h}\frac{\partial q(v(t))}{\partial v(t)} + \frac{\partial i(v(t))}{\partial v(t)}. \qquad (21)$$

Solving (19) yields an approximation to $v(t_0+h) = \phi(v(t_0), t_0, t_0+h)$. Implicitly differentiating (19) for $v(t_0+h)$ with respect to $v(t_0)$ yields

$$J_f(v(t_0+h))\frac{\partial v(t_0+h)}{\partial v(t_0)} = \frac{1}{h}\frac{\partial q(v(t_0))}{\partial v(t_0)}\frac{\partial v(t_0)}{\partial v(t_0)}. \qquad (22)$$

Note here that $\frac{\partial v(t_0)}{\partial v(t_0)} \in \Re^{N\times N}$ is the identity matrix and in general $\frac{\partial v(t_0+h)}{\partial v(t_0)} \in \Re^{N\times N}$ is not the identity.

Given $v(t_0)$, (19) can be repeatedly applied to find $v(t_0+T) = \phi(v(t_0), t_0, t_0+T)$, and (22) can be repeatedly applied to find the sensitivity matrix $\partial v(t_0+T)/\partial v(t_0) = \partial\phi(v(t_0), t_0, t_0+T)/\partial v(t_0)$. Note that J_f is required in both (20) and (22), and thus the sensitivity matrix update can be made more efficient by factoring J_f once and using it for both computations. However, the sensitivity matrix is still expensive to compute, because it is an $N \times N$ full matrix. We return to this point at the end of section 4.

4 Implementation in *Nitswit*

Both the classical direct methods and the mixed frequency-time methods have been implemented in the simulation program *Nitswit*, which is written in the computer language "C." *Nitswit* takes as input a file with a SPICE-like description of the circuit, that is, a list of elements (MOS transistors, resistors, capacitors, etc) with their node connections, and a list of options to select among methods. If the mixed frequency-time method is used, a switching clock period and one or two input frequencies (two for intermodulation distortion) along with a number of harmonics must be specified. The program produces some form of transient waveforms and Fourier series coefficients, depending on the options selected.

4.1 Application Examples

To demonstrate the effectiveness of the algorithms used in the *Nitswit* program, we consider analyzing the distortion of a switched-capacitor low pass filter. Nitswit is particularly efficient for switched-capacitor filters for several reasons. First, a switched-capacitor filters is usually followed by a sampler, and so only

the initial point of each cycle is needed. Second, the circuits are designed so that the distortion present in the sequence of initial points is small, so if driven by a sinusoid, only a few harmonics are significant and only a few clock cycles need to be computed. Finally, the state transition function for a switched-capacitor filter over a clock cycle is nearly affine (linear plus a constant), and therefore the Newton method in (17) converges in just a few iterations.

Tables 1 and 2 show the results when *Nitswit* is used to compute the distortion produced by a single-pole switched-capacitor low-pass filter with a clock of 500KHz, a bandwidth of 30KHz, and an input frequency of 20KHz. To simulate the effect of nonlinear filter capacitance on distortion, the filter capacitors are assumed to be first order voltage-controlled nonlinear capacitors with capacitance $c = c_0(1 + \alpha v_c)$. In Table 1, the distortion, specifically the magnitude of the first two harmonics, is given for several different values of α. The distortion for the same low-pass filter circuit with linear filter capacitors is given as a function of the clock rise and fall time in Table 2.

α	Mag. First Harm.	Mag. Sec. Harm.
0.001	0.00057	0.00010
0.01	0.0014	0.00007
0.1	0.0101	0.00024

Table 1. *Nitswit* Relative distortion results for a switched-capacitor low-pass filter as a function of increasing filter capacitor nonlinearity.

Clock	Mag. First Harm.	Mag. Sec. Harm.
1ns	0.000020	5.99e-7
10ns	0.00020	0.000037
100ns	0.0018	0.000054

Table 2. *Nitswit* distortion results for a switched-capacitor low-pass filter as a function of increasing clock rise and fall time.

4.2 Comparison to Direct Methods

The program *Nitswit* contains two algorithms capable of finding the steady-state response of a circuit. The first is simply a transient analysis that continues until a steady-state is achieved. The second, of course, is the mixed frequency-time algorithm. Coding both algorithms into the same simulator provides a fair evaluation of the mixed frequency-time approach.

Results for five circuits are given in Table 3 below. The first, *sclpf*, is an RC one-pole SC filter. The second, *scop*, is a one-pole active CMOS low pass filter. The last circuit, *mixer*, is a double-balanced switching mixer with a 1.001 MHz rf and a 1MHz input.The last, *frog*, is a five pole Chebyshev active CMOS leap frog filter with 0.1dB ripple. This circuit is driven with a 1MHz clock, has a 20kHz bandwidth, and is being driven with a 1kHz test signal to measure its distortion.

circuit			direct	mixed frequency-time		
name	nodes	cycles/ period	time (sec)	harmonics, cycles	Newton iterations	time (sec)
sclpf	2	33	24.5	3,7	3	4.3
scop	13	100	522	3,7	6	90
mixer	34	1000	7132	3,7	4	161
frog	77	1000	12,987	3,7	6	1228

Table 3. *Nitswit* results from a VAX 8650 running ULTRIX 2.0.

Examination of the results above indicate as much as an order of magnitude speed increase over traditional methods, but this is not as much as one would expect. Much of the CPU time for large circuits, such as *frog*, is spent calculating the dense sensitivity matrix and factoring the Jacobian in (18). It does turn out however, that almost all the entries of the sensitivity matrix are near zero, and this suggests significant speed improvements can be achieved by ignoring those terms. In addition, we expect to get improved performance by switching

to relaxation techniques to solve (14). Preliminary experiments indicate the relaxation converges quickly and reliably, and is much faster than sparse LU factorization.

5 Conclusion

A new efficient mixed frequency-time approach to computing steady-state and intermodulation distortion of switching filters without resorting to macromodeling or the slow-clock approach has been presented. The method works by computing the solution to the differential equation system associated with a circuit for only J clock cycles, where J is the number coefficients needed in the Fourier series to represent accurately the sequence of initial points in each clock cycle. Thus, this method is particularly efficient when the number of coefficients in the Fourier series is many fewer than the number of clock cycles in one input signal period.

Since our approach finds the steady-state solution directly and performs a circuit-level simulation, it is capable of accurately predicting distortion performance. This mixed frequency-time approach can also be used when the input consists of the sum of two periodic signals at unrelated frequencies. Thus, the intermodulation distortion can be directly computed, which is particularly useful for bandpass filters. Also, the fact that steady-state is computed directly implies an additional advantage over transient methods when high-Q filters are simulated. One final point, the mixed frequency-time method can also be adapted to the macromodeling approach used in other switching filter simulators, accelerating those methods as well when the steady-state solution is desired.

Future work on this method will be to adapt it to other traditionally hard-to-simulate circuits like switching power supplies and phase-locked loops. Another

important aspect of this algorithm is that, upon examination of (14), it is clear that the J integrations of the differential equation to compute the J ϕ's and their derivatives are independent. The other step, solving the sparse matrix problem in (17), seems, as mentioned above, to be very amenable to solution by relaxation. Therefore, the mixed frequency-time algorithm is extremely well suited to implementation on a parallel processor.

5.1 Acknowledgements

We would like to acknowledge the discussions with Professor A. R. Newton; the help of Robert Armstrong, Andrew Lumsdaine, Keith Nabors, Mark Reichelt, and Hormoz Yaghutiel; and the support of Barbara Bratzel and Mary Glanville. In addition, we would especially like to thank Res Saleh, for his many contributions to this project. This work was supported by the Defense Advanced Research Projects Agency contract N00014-87-K-825, Hewlett-Packard, and the California MICRO program.

References

[chua81] L. Chua and A. Ushida. "Algorithms for computing almost-periodic steady-state response of nonlinear systems to multple input frequencies," *IEEE Transactions on Circuits and Systems*, vol. CAS-28, pp. 953-971, October 1981.

[deman80] H. De Man, J. Rabaey, G. Arnout, J. Vandewalle. "Practical implementation of a general computer aided design technique for switched capacitor circuits," *IEEE Journal of Solid-State Circuits*, vol. SC-15, pp. 190-200, April 1980.

[kundert88a] K. S. Kundert, G. B. Sorkin, A. Sangiovanni-Vincentelli. "Applying harmonic balance to almost-periodic circuits," *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-36, February 1988.

[nagel75] L. W. Nagel. *SPICE2: A Computer Program to Simulate Semiconductor Circuits*. Electronics Research Lab Report, ERL M520, Univ. of Calif., Berkeley, May 1975.

[trick75] T. N. Trick, F. R. Colon, S. P. Fan. "Computation of capacitor voltage and inductor currrent sensitivities with respect to initial conditions for the steady state analysis of nonlinear periodic circuits," *IEEE Transactions on Circuits and Systems*, vol. CAS-22, pp. 391-396, May 1975.

[tsividis79] Y. P. Tsividis. "Analysis of switched capacitor networks," *IEEE Transactions on Circuits and Systems*, vol. CAS-26, pp. 935-946, November 1979.

[weeks73] W. T. Weeks, A. J. Jimenez, G. W. Mahoney, D. Mehta, H. Quasemzadeh, T. R. Scott. "Algorithms for ASTAP — A Network Analysis Program," *IEEE Transactions on Circuit Theory*, pp. 628-634, Nov. 1973.

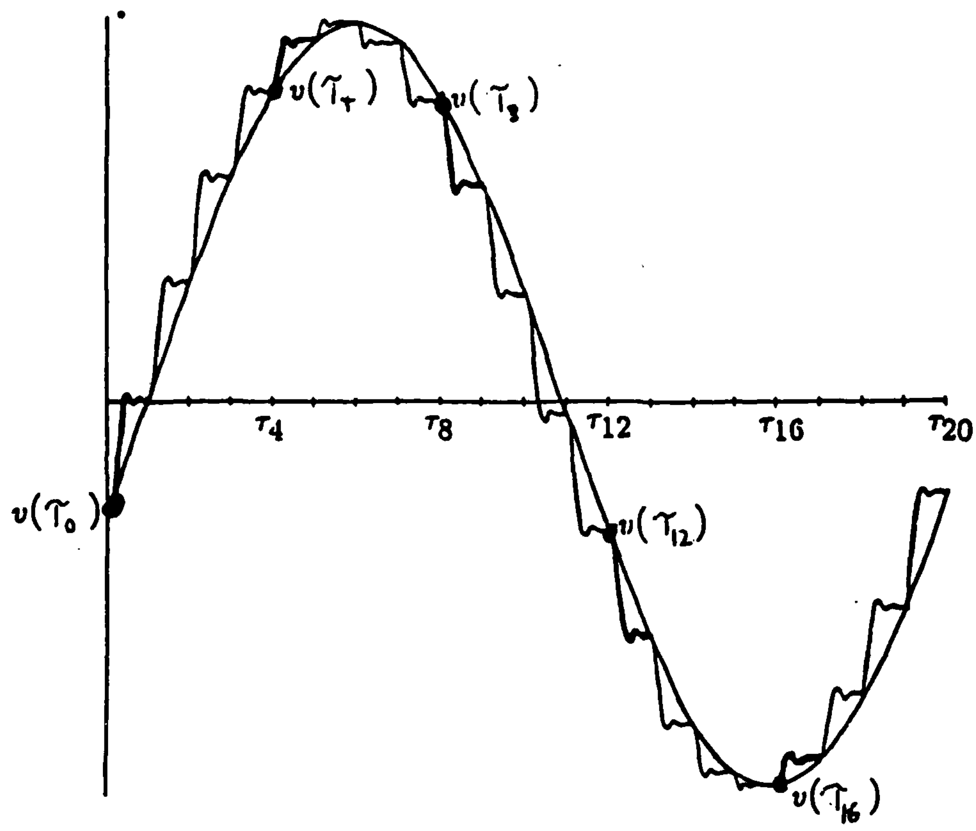

Figure 1: The response of a switching filter circuit to a periodic function, with the initial points of each cycle denoted.

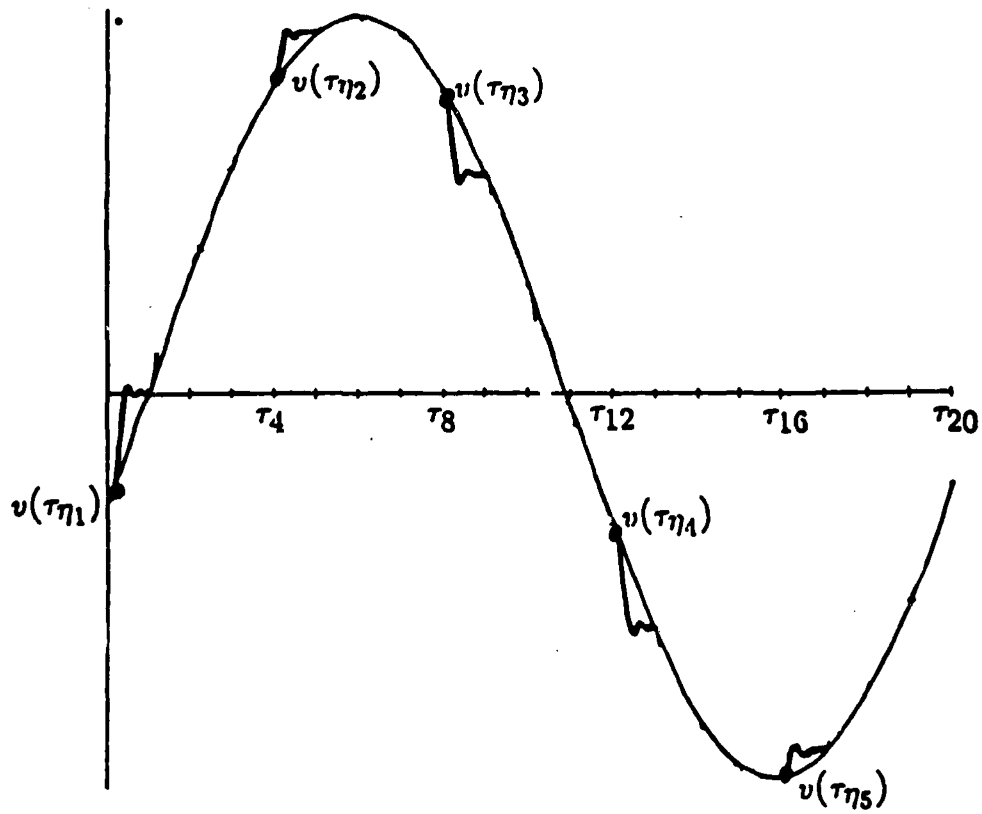

Figure 2: The response of a switching filter circuit and the periodic function of the initial points. The J cycles used in the calculation are emphasized.